

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВОЛИНСЬКІЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЛЕСІ УКРАЇНКИ**

**Кафедра загальної математики та методики навчання інформатики**

На правах рукопису

**ВІРНИЙ НІКІТА ВАДИМОВИЧ  
СУЧАСНІ ТЕХНОЛОГІЇ НАВЧАННЯ ВЕБ-ПРОГРАМУВАННЯ У  
СТАРШІЙ ТА ПРОФІЛЬНІЙ ШКОЛІ**

Спеціальність: 014 «Середня освіта (Інформатика)»

Освітньо-професійна програма Середня освіта. Інформатика

Робота на здобуття освітнього ступеня «Магістр»

Науковий керівник:

**Собчук Оксана Миколаївна**

кандидат педагогічних наук,

доцент кафедри загальної математики

та методики навчання інформатики

**РЕКОМЕНДОВАНО ДО ЗАХИСТУ**

Протокол № \_\_\_\_\_

засідання кафедри загальної математики

та методи навчання інформатики від

\_\_\_\_\_ 2024 року

Завідувач кафедри:

\_\_\_\_\_

**ЛУЦЬК – 2024**

## Анотація

**Вірний Н. В. Методика викладання веб-програмування у старшій та профільній школі – Рукопис**

Випускна кваліфікаційна робота за спеціальністю 014 Середня освіта (Інформатика). – Волинський національний університет імені Лесі Українки. Луцьк, 2024.

Кваліфікаційна робота присвячена дослідженню сучасних методик викладання веб-програмування у старшій та профільній школі. У першому розділі аналізуються основні концепції веб-програмування, його значення у сучасному суспільстві, а також роль у навчальних програмах. Проведено порівняння зарубіжного та вітчизняного досвіду викладання веб-програмування.

Другий розділ зосереджений на використанні інтерактивних платформ, хмарних сервісів та інноваційних технологій для викладання веб-програмування. Розглянуто можливості впровадження візуалізацій, симуляторів та інтеграції STEAM-підходу в навчальний процес.

У другому розділі також запропоновано структуру уроку веб-програмування, яка включає теоретичну та практичну складові. Висвітлено методи залучення учнів до проєктної діяльності, гейміфікації та використання реальних кейсів для підвищення мотивації.

Результати дослідження підтвердили ефективність запропонованої методики у формуванні практичних навичок та підвищенні мотивації учнів до вивчення веб-програмування. Рекомендації щодо впровадження можуть бути використані для удосконалення освітніх програм у середній школі.

**Ключові слова:** веб-програмування, інтерактивні платформи, STEAM-підхід, гейміфікація, навчальний процес, проєктна діяльність, сучасні технології.

## **Annotation**

### **Virnyi N.V. Methodology for Teaching Web Programming in High and Specialized Schools – Manuscript**

Graduate Qualification Work in Specialty 014 Secondary Education (Informatics). – Lesya Ukrainka Volyn National University. Lutsk, 2024.

This qualification work focuses on the research of modern methodologies for teaching web programming in high and specialized schools. The first chapter analyzes the fundamental concepts of web programming, its role in modern society, and its place in educational curricula. A comparison of international and domestic teaching practices is provided.

The second chapter explores the use of interactive platforms, cloud services, and innovative technologies for teaching web programming. It examines opportunities for incorporating visualizations, simulators, and the integration of the STEAM approach into the educational process.

The second chapter presents a lesson structure for web programming, comprising theoretical and practical components. Methods for engaging students in project-based activities, gamification, and using real-world cases to enhance motivation are highlighted.

The research results confirmed the effectiveness of the proposed methodology in developing practical skills and increasing students' motivation to study web programming. The recommendations can be utilized to improve educational programs in secondary schools.

**Keywords:** web programming, interactive platforms, STEAM approach, gamification, educational process, project-based learning, modern technologies.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>4</b>
<b>РОЗДІЛ 1: ТЕОРЕТИЧНІ ОСНОВИ НАВЧАННЯ</b>	
<b>ВЕБ-ПРОГРАМУВАННЯ.....</b>	<b>6</b>
1.1. Поняття веб-програмування та його роль у сучасному світі.....	6
1.2. Місце веб-програмування в освітній програмі старшої та профільної школи.....	8
1.3. Основні принципи та вимоги до викладання веб-програмування.....	9
1.4. Аналіз сучасних освітніх стандартів і вимог до викладання веб-програмування: вітчизняний та зарубіжний досвід.....	10
<b>РОЗДІЛ 2: МОЖЛИВОСТІ ВИКОРИСТАННЯ СУЧАСНИХ ТЕХНОЛОГІЙ</b>	
<b>У НАВЧАННІ ВЕБ-ПРОГРАМУВАННЯ У СТАРШІЙ ТА ПРОФІЛЬНІЙ ШКОЛІ.....</b>	<b>17</b>
2.1. Сучасні технології у навчанні веб-програмування.....	17
2.1.1 Використання інтерактивних платформ і середовищ для навчання веб-програмування.....	17
2.1.2. Хмарні сервіси та онлайн-інструменти у викладанні веб-програмування.....	19
2.1.3. Використання візуалізацій та симуляторів у навчальному процесі.	20
2.1.4. Роль технологій дистанційного навчання у формуванні навичок веб-програмування.....	22
2.1.5. Інтеграція STEAM-підходу у викладання веб-програмування.....	25
2.1.6. Інтерактивні методи навчання: гейміфікація, квести, вебінари.....	27
2.2. Проектне навчання як основа для формування навичок веб-програмування.....	30
2.3. Залучення реальних кейсів та проектів для навчання веб-програмування..	32
2.4. Розробка структури уроку веб-програмування (теоретична і практична частини).....	33
2.5. Аналіз ефективності розробленої методики.....	36
2.5.1. Організація експериментального дослідження ефективності технологій навчання.....	36
2.5.2. Оцінка рівня сформованості знань і навичок у старшокласників....	38
2.5.3. Порівняльний аналіз результатів до і після впровадження методики..	39
2.5.4. Вплив сучасних технологій на мотивацію та компетентності учнів....	41
<b>ВИСНОВКИ.....</b>	<b>43</b>
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....</b>	<b>47</b>

**ДОДАТОК 1.....51**

## ВСТУП

Сучасний світ розвивається в умовах стрімкого прогресу інформаційних технологій, які стають невід'ємною частиною різних сфер діяльності людини. Однією з ключових навичок, що забезпечують успішність у технологічному суспільстві, є вміння працювати з веб-технологіями. У зв'язку з цим навчання веб-програмуванню набуває особливого значення, оскільки дозволяє не лише зрозуміти основи сучасних інформаційних технологій, а й розвивати критичне мислення, креативність та навички вирішення проблем.

**Тема дипломної роботи** є надзвичайно актуальною, адже сьогодні веб-програмування є основою створення цифрового контенту та управління інформацією в Інтернеті. Особливо важливим стає забезпечення якісного викладання цих знань у старшій та профільній школі, адже саме тут формуються професійні інтереси учнів і базові компетенції для майбутнього кар'єрного розвитку.

**Мета роботи** полягає в аналізі сучасних технологій, які можуть бути використані для навчання веб-програмування у старшій та профільній школі, та розробці ефективної методики їх впровадження.

### **Основними завданнями дослідження є:**

Аналіз теоретичних основ навчання веб-програмування, його значення в освітньому процесі.

Вивчення сучасних освітніх технологій, що можуть бути застосовані для викладання веб-програмування.

Розробка методичних рекомендацій для викладачів щодо ефективного навчання веб-програмування.

Проведення експериментального дослідження ефективності запропонованих підходів і оцінка їх впливу на навчальний процес.

**Об'єкт дослідження** – процес навчання веб-програмування у старшій та профільній школі.

**Предмет дослідження** – сучасні технології та методи, що застосовуються для викладання веб-програмування.

Методи дослідження:

- Теоретичні – аналіз літератури, вивчення освітніх стандартів і програм.
- Практичні – спостереження за навчальним процесом, проведення експерименту, анкетування учнів і викладачів.
- Статистичні – обробка даних, отриманих під час експерименту.

Важливість дослідження зумовлена необхідністю адаптації освіти до викликів цифрової епохи, створення ефективних умов для формування компетентностей учнів у сфері веб-програмування та підвищення їхньої конкурентоспроможності на ринку праці.

Практичне значення роботи полягає у створенні рекомендацій для викладачів і навчальних закладів, які прагнуть запровадити сучасні технології навчання веб-програмуванню. Запропоновані підходи можуть бути використані для розробки навчальних планів, проведення уроків та позакласних заходів.

Ця робота спрямована на розробку практичних рішень, що допоможуть удосконалити процес навчання веб-програмуванню, зробивши його більш ефективним, цікавим та доступним для сучасного учня.

## **РОЗДІЛ 1: ТЕОРЕТИЧНІ ОСНОВИ НАВЧАННЯ ВЕБ-ПРОГРАМУВАННЯ**

### **1.1. Поняття веб-програмування та його роль у сучасному світі**

Веб-програмування – це процес створення та підтримки веб-додатків і сайтів, які є основою функціонування Інтернету. Воно поєднує в собі використання різних мов програмування, технологій і інструментів для розробки інтерфейсів (фронтенд) і функціональної частини серверів (бекенд). Основними мовами веб-програмування є HTML, CSS, JavaScript, PHP, Python, Ruby та інші.

На базовому рівні веб-програмування можна уявити як процес створення структури веб-сторінки (HTML), її оформлення (CSS) та забезпечення інтерактивності (JavaScript). З часом, в умовах зростання потреб бізнесу та споживачів, веб-технології перетворилися на потужний інструмент для створення складних інформаційних систем, таких як соціальні мережі, електронні магазини, онлайн-банкінг тощо.

#### **Веб-програмування в сучасному суспільстві**

Роль веб-програмування у сучасному світі важко переоцінити, адже воно забезпечує взаємодію людей, компаній та урядів через Інтернет. Завдяки веб-технологіям сьогодні можна виконувати різноманітні завдання:

- Отримувати інформацію за лічені секунди;
- Здійснювати покупки в онлайн-магазинах;
- Спілкуватися через соціальні мережі та месенджери;
- Організовувати дистанційне навчання та роботу.

За даними звіту Digital 2023, близько 63% світового населення користується Інтернетом, що свідчить про глобальний характер впливу веб-технологій на всі аспекти життя. Саме веб-програмування є основою для створення платформ, які формують цей цифровий простір.

#### **Сфери застосування веб-програмування**

Освіта. Розробка платформ для дистанційного навчання, таких як Google Classroom, Moodle, допомагає інтегрувати новітні технології у процес здобуття знань.

Бізнес. Сайти та веб-додатки стали основними каналами комунікації між компаніями та клієнтами. Електронна комерція та автоматизація бізнес-процесів є важливими елементами сучасної економіки.

Соціальна взаємодія. Соціальні мережі, наприклад Facebook, Instagram, Twitter, стали основними платформами для спілкування та обміну інформацією.

Технології майбутнього. З появою концепції Web 3.0, розробники почали працювати над децентралізованими додатками та використанням блокчейну, що відкриває нові можливості для створення безпечного та прозорого цифрового середовища.

#### Роль веб-програмування в освіті

Окрім його широкого використання в різних сферах, веб-програмування відіграє важливу роль у підготовці молоді до сучасних викликів. Навчання веб-програмуванню розвиває у школярів такі компетенції, як:

- Логічне мислення;
- Навички роботи в команді через спільні проєкти;
- Здатність до креативного вирішення задач.

Сучасна освіта активно інтегрує веб-технології, що створює додаткові можливості для учнів. За результатами дослідження, опублікованого у 2022 році на платформі Springer, студенти, які опановують основи програмування ще у школі, демонструють вищий рівень успішності у технологічних спеціальностях.

Веб-програмування стало невід'ємною частиною нашого життя. Від його розвитку залежить зручність, безпека та інноваційність сучасних технологій. Зважаючи на швидкий темп змін, навчання веб-програмуванню в школі є важливим інструментом підготовки молодого покоління до життя в цифровому світі.

## **1.2. Місце веб-програмування в освітній програмі старшої та профільної школи**

Веб-програмування займає важливе місце в освітніх програмах старшої та профільної школи. В умовах швидкого розвитку цифрових технологій та значного впливу Інтернету на всі сфери життя, освіта повинна відповідати новим вимогам, що ставляться перед сучасним суспільством. Тому вивчення веб-програмування не лише надає учням практичні навички для вирішення реальних завдань, але й розвиває їхні здатності до критичного мислення, аналітичних роздумів та вирішення комплексних проблем.

Веб-програмування дозволяє учням старших класів отримати не лише базові знання у сфері інформаційних технологій, але й глибше освоїти інструменти створення веб-сайтів, роботи з базами даних, створення інтерфейсів та інтерактивних додатків. Ось чому воно має стати важливою складовою частиною навчальних програм, зокрема в рамках профільних класів.

Крім того, навчання веб-програмуванню сприяє розвитку багатьох важливих компетенцій. Наприклад, програмування допомагає школярам мислити логічно і системно, що є важливим у багатьох сферах діяльності. Вивчення створення сайтів і додатків розвиває їхню креативність, оскільки цей процес потребує творчого підходу до вирішення задач, пошуку оригінальних рішень для взаємодії з користувачем. Також важливим аспектом є розвиток командних навичок, адже в реальному світі багато проєктів, у тому числі веб-розробки, реалізуються колективно, що вимагає взаємодії, комунікації та здатності працювати в групі.

Веб-програмування також забезпечує учнів знаннями, які відповідають вимогам ринку праці, адже сучасні технології все більше визначають розвиток різних сфер економіки. За даними звіту Європейської Комісії "Digital Education Action Plan 2021-2027", більше 80% шкіл в ЄС уже впроваджують базові курси програмування в освітні програми. Україна також активно рухається в напрямку

інтеграції технологій у шкільну освіту, що є важливим кроком на шляху до підготовки конкурентоспроможних фахівців у галузі ІТ. [5]

У рамках профільного навчання, яке впроваджується в старших класах, веб-програмування може бути як частиною основної програми, так і вибіркоким курсом. Основні теми, які охоплює курс програмування, включають основи створення веб-сайтів, роботу з різними мовами програмування (HTML, CSS, JavaScript), основи безпеки в Інтернеті, а також використання баз даних для створення динамічних веб-ресурсів.

Інтеграція таких курсів у шкільну програму не лише забезпечує учнів важливими технічними знаннями, але й готує їх до викликів сучасного світу, в якому інформаційні технології все більше стають основою для розвитку бізнесу, науки та суспільства в цілому. У цьому контексті важливою перевагою є і міждисциплінарний підхід, адже навчання веб-програмуванню включає не лише інформатику, але й математичні, логічні та навіть художні аспекти. Наприклад, для розробки інтерфейсів веб-сайтів потрібні знання з дизайну, що сприяє розвитку естетичного сприйняття учнів.

Таким чином, веб-програмування є ключовим елементом освітніх програм старшої та профільної школи. Воно надає учням важливі навички, які будуть корисні в їхній майбутній кар'єрі, а також сприяє розвитку критичного мислення, креативності та інших цінних компетенцій, які важливі не лише в ІТ-сфері, а й у багатьох інших сферах життя.

### **1.3. Основні принципи та вимоги до викладання веб-програмування.**

Викладання веб-програмування вимагає дотримання ряду принципів, які забезпечують ефективне засвоєння знань і розвиток практичних навичок. Насамперед, воно повинно бути орієнтованим на формування компетенцій, які відповідають сучасним вимогам ІТ-сфери. Уроки мають не лише передавати знання, але й навчати застосовувати їх у реальних умовах.

Основоположним принципом є практична спрямованість. Учні повинні мати змогу реалізувати знання з HTML, CSS, JavaScript та інших технологій на

практиці, створюючи реальні веб-проекти. Ця діяльність дозволяє сформувавши розуміння зв'язку між теорією та її застосуванням. Важливим є також принцип інтерактивності. Залучення учнів до активної роботи на уроці, взаємодія в командах і використання цифрових платформ сприяють більш глибокому зануренню в тему.

Викладання веб-програмування має базуватися на принципі доступності матеріалу. Навчальний контент необхідно адаптувати до рівня знань і вікових особливостей учнів. Складні теми слід подавати через зрозумілі приклади та пояснення, використовуючи сучасні навчальні інструменти, такі як візуалізації чи інтерактивні симулятори.

Іншим важливим принципом є актуальність. Оскільки веб-технології швидко розвиваються, програма навчання повинна враховувати сучасні тренди та оновлення в сфері програмування. Учні мають знайомитися з інструментами, які активно використовуються в індустрії. Це мотивує їх до навчання та дозволяє їм бути конкурентоспроможними на ринку праці в майбутньому.

Також викладання веб-програмування вимагає інтеграції міжпредметних зв'язків. Поєднання інформатики, математики, англійської мови (оскільки велика кількість документації доступна англійською), а також інших дисциплін допомагає створити цілісну картину навчання. Такий підхід розвиває критичне мислення, аналітичні здібності та креативність.

На завершення, ключовим елементом викладання є безперервний зворотний зв'язок. Викладачі мають оцінювати результати навчання, виявляти прогалини в знаннях і коригувати методику викладання. Це допомагає не лише учням краще засвоїти матеріал, але й забезпечує відповідність навчання сучасним освітнім стандартам.

#### **1.4. Аналіз сучасних освітніх стандартів і вимог до викладання веб-програмування: вітчизняний та зарубіжний досвід**

Веб-програмування, як одна з ключових складових інформаційних технологій, набуває особливого значення в освітніх програмах, зокрема в

старшій та профільній школі. Враховуючи швидкий розвиток технологій та їхній вплив на всі сфери життя, виникає необхідність у розробці чітких освітніх стандартів і вимог, які б забезпечили належний рівень підготовки учнів у сфері веб-розробки.

### Сучасні освітні стандарти в Україні

В Україні основним документом, який регламентує викладання інформаційних технологій у школах, є "Державний стандарт базової і повної середньої освіти", затверджений Міністерством освіти і науки України. Відповідно до цього стандарту, вивчення інформаційних технологій є важливою складовою навчального плану, особливо в профільних класах, де спеціалізація в галузі комп'ютерних наук або технічних дисциплін дозволяє учням здобувати поглиблені знання та навички.

У рамках загальної освіти навчання веб-програмуванню передбачає вивчення базових технологій, таких як HTML, CSS, JavaScript, а також основ програмування. Водночас викладання повинно відповідати вимогам, які ставляться до розвитку ключових компетенцій учнів, включаючи:

- Цифрову грамотність — здатність учнів не лише користуватися комп'ютерами та інтернетом, а й створювати програмні продукти.
- Критичне мислення — вміння аналізувати і вирішувати проблеми за допомогою програмування та сучасних технологій.
- Проблемно-орієнтований підхід — навчання на основі реальних завдань та кейсів, що дозволяє учням застосовувати набуті знання в практичній діяльності.

Згідно з вимогами державного стандарту, в школах має бути забезпечений доступ до сучасних цифрових інструментів і ресурсів для викладання предмету. У результаті цього учні можуть використовувати різні платформи для навчання веб-програмуванню, працювати з інтерактивними середовищами, що дозволяють розвивати навички розробки веб-сайтів, мобільних додатків, а також вивчати принципи оптимізації веб-ресурсів.

Європейські та міжнародні стандарти викладання веб-програмування

Міжнародний досвід вказує на важливість інтеграції програмування в освітні програми. Так, в рамках Європейського Союзу було розроблено низку ініціатив, спрямованих на популяризацію цифрових технологій серед молоді. Зокрема, в програмі "Digital Education Action Plan 2021-2027" зазначено, що країни-члени ЄС повинні забезпечити учням доступ до якісної освіти в галузі цифрових технологій, включаючи веб-програмування, що стає основою для розвитку інноваційних компетенцій. [13]

Відповідно до цієї програми, освітні стандарти в Європі охоплюють такі аспекти:

Основи програмування — курс з основ програмування, який вивчається в школах для забезпечення базового рівня цифрової грамотності серед учнів.

Інтердисциплінарний підхід — веб-програмування інтегрується з іншими навчальними дисциплінами, такими як математика, фізика, мистецтво, щоб учні могли бачити застосування технологій у різних сферах.

Проектний підхід — навчання веб-програмуванню має на меті розвиток проектних навичок учнів, тобто вони повинні створювати реальні проекти, що дозволяє їм застосовувати набуті знання на практиці.

Вимоги до викладачів і методики навчання

Однією з основних вимог до викладання веб-програмування є наявність висококваліфікованих викладачів, які мають відповідну підготовку в галузі інформаційних технологій. Освітні стандарти вимагають, щоб учителі мали глибокі знання в галузі програмування, володіли сучасними інструментами та методами викладання, що забезпечують ефективне навчання.

Методика викладання веб-програмування передбачає поєднання теоретичних та практичних занять. Важливим аспектом є використання інтерактивних ресурсів, таких як онлайн-курси, платформи для самостійного навчання, а також програмні засоби, що дозволяють учням практикуватися в створенні веб-сайтів і додатків. Веб-програмування повинно бути не лише інструментом для освоєння технічних навичок, але й способом для розвитку

креативності, комунікативних навичок і вміння працювати в команді, що є важливими аспектами сучасної освіти.

### **Виклики і проблеми в сучасному викладанні веб-програмування**

Незважаючи на значний прогрес в інтеграції веб-програмування в освітні програми, існують певні виклики, з якими стикаються школи. Одним з основних є недостатня технічна оснащеність шкіл, що не дозволяє повною мірою використовувати сучасні платформи і інструменти для навчання. Іншим викликом є недостатня підготовка педагогічного складу, адже для викладання веб-програмування потрібні фахівці, які мають не лише педагогічні навички, а й глибокі знання в галузі ІТ.

Таким чином, сучасні освітні стандарти та вимоги до викладання веб-програмування орієнтовані на інтеграцію цієї дисципліни в загальну систему освіти з урахуванням новітніх технологій та вимог ринку праці. Вони передбачають як розвиток базових технічних навичок, так і формування критичного мислення, інноваційного підходу до вирішення проблем та вміння працювати в команді. Задля успішної реалізації цих стандартів необхідно постійно вдосконалювати методику викладання та підтримувати учителів на всіх етапах професійного розвитку.

Викладання веб-програмування, будучи важливою частиною навчального процесу у профільній школі, істотно відрізняється в Україні та за кордоном. Різниця у підходах зумовлена освітніми стандартами, технічним забезпеченням, кваліфікацією викладачів і культурними особливостями. Розглянемо основні аспекти цих відмінностей і спільних рис.

### **Зарубіжний досвід викладання веб-програмування**

#### **Освітні програми та стандарти**

У багатьох розвинених країнах веб-програмування є інтегрованою частиною освітньої програми з ранніх етапів навчання. Наприклад, у Великій Британії курс "Computing" у шкільній програмі охоплює основи програмування,

включаючи веб-розробку. У США школи використовують платформу Code.org і її проєкти, які навчають дітей основ програмування через інтерактивні вправи та проєкти.

Європейський підхід також передбачає використання міжнародних програм сертифікації, таких як ECDL (European Computer Driving License), де веб-програмування входить до розділів із сучасних цифрових компетенцій.

#### Методика навчання

Зарубіжна методика базується на інноваційних підходах:

- Проєктне навчання: студенти працюють над реальними проєктами, такими як створення веб-сайтів чи додатків.
- Ігрові методи: навчальні платформи типу \*Scratch\* або \*Khan Academy\* активно використовують ігрову механіку для залучення учнів.
- Рання спеціалізація: курси з веб-програмування пропонуються вже у середній школі, що дозволяє школярам визначитися з майбутньою професією ще до закінчення школи.

#### Технічна база

У багатьох країнах Європи, а також у США та Канаді, школи забезпечені сучасною технікою: ноутбуками, планшетами та швидкісним доступом до інтернету. Крім того, багато навчальних закладів співпрацюють із ІТ-компаніями, що забезпечують доступ до професійного програмного забезпечення.

### **Вітчизняний досвід викладання веб-програмування**

#### Освітні програми та стандарти

В Україні веб-програмування часто є частиною факультативних курсів або викладається в рамках інформатики у профільних класах. Зміст навчання нерідко обмежується базовими темами (HTML і CSS), тоді як більш складні аспекти, такі як JavaScript чи основи роботи з базами даних, можуть бути представлені лише у спеціалізованих школах чи гуртках.

Оновлений державний стандарт передбачає інтеграцію ІТ-компетенцій у навчальні програми, але через брак спеціалістів і ресурсів впровадження таких ініціатив іде повільно.

#### Методика навчання

В Україні акцент робиться на теоретичній підготовці, у той час як практична частина викладання часто залишається обмеженою. Хоча дедалі більше шкіл починають використовувати інтерактивні ресурси (наприклад, Prometheus, EdEra), цей процес ще не став масовим. Крім того, багато учнів звертаються до позашкільних онлайн-курсів. [16]

#### Технічна база

Багато українських шкіл стикаються з проблемами недостатнього технічного забезпечення. Старі комп'ютери, слабкий доступ до інтернету та обмежений вибір програмного забезпечення є основними викликами.

#### Спільні риси та відмінності

Аспект	Зарубіжний досвід	Український досвід
Освітні програми	Обов'язкове включення до курсу інформатики, рання спеціалізація.	Факультативи або частина курсу інформатики.
Методика	Проектне навчання, інтерактивні платформи, ігрові методи.	Теоретичний підхід, обмежена практика.
Технічна база	Сучасні технології, співпраця з ІТ-компаніями.	Недостатнє технічне забезпечення.

Україні необхідно орієнтуватися на найкращі практики зарубіжних країн у викладанні веб-програмування, інтегруючи інноваційні методи та забезпечуючи школи сучасними ресурсами. Зокрема, важливими напрямками для вдосконалення є:

- посилення практичної складової навчання;
- створення партнерств із компаніями для оновлення технічної бази;
- розробка національних програм підтримки викладачів для підвищення їхньої кваліфікації.

Система освіти в Україні має значний потенціал, і впровадження найкращих зарубіжних практик допоможе створити покоління учнів, готових до викликів сучасного ІТ-світу.

## РОЗДІЛ 2: МОЖЛИВОСТІ ВИКОРИСТАННЯ СУЧАСНИХ ТЕХНОЛОГІЙ У НАВЧАННІ ВЕБ-ПРОГРАМУВАННЮ У СТАРШІЙ ТА ПРОФІЛЬНІЙ ШКОЛІ

### 2.1. Сучасні технології у навчанні веб-програмування

#### 2.1.1 Використання інтерактивних платформ і середовищ для навчання веб-програмування.

Інтерактивні платформи та середовища для навчання веб-програмування стали невіддільною частиною сучасного освітнього процесу. Їх використання базується на ідеї поєднання теорії з практикою, доступності матеріалів і залученні учнів через інтерактивні завдання. Такі ресурси сприяють тому, що учні можуть не лише отримувати знання, а й одразу застосовувати їх на практиці, що робить процес навчання більш цікавим і ефективним.

Головною перевагою інтерактивних платформ є їхня адаптивність. Учні можуть навчатися у власному темпі, зосереджуючись на тих аспектах, які потребують більшої уваги. Наприклад, платформи на кшталт Codecademy або freeCodeCamp пропонують структуровані курси, що включають теоретичні матеріали, інтерактивні вправи й реальні практичні проекти. Це дозволяє учням не лише опанувати основи веб-програмування, а й створити власне портфоліо для подальшого використання в навчанні чи роботі.

Іншою важливою характеристикою інтерактивних середовищ є практична орієнтованість. Учні можуть писати код безпосередньо на платформі, тестувати його та миттєво бачити результати своєї роботи. Це не лише полегшує процес засвоєння знань, а й сприяє розвитку навичок вирішення реальних задач, які є ключовими у професії веб-розробника. Платформи також надають зворотний зв'язок, що дозволяє учням швидко виправляти помилки й удосконалювати свої навички. [10]

До найпопулярніших інтерактивних ресурсів належать такі платформи, як freeCodeCamp, Codecademy, Khan Academy та W3Schools. Вони відрізняються зручним інтерфейсом, різноманітними навчальними матеріалами та можливістю поступового освоєння знань, починаючи від базових принципів HTML і CSS до роботи з JavaScript та іншими складними технологіями. Наприклад, на платформі freeCodeCamp учні мають змогу працювати над проєктами, які відповідають сучасним вимогам індустрії, а також отримувати сертифікати після завершення курсів.

Інтеграція таких платформ у навчальний процес старшої школи відкриває широкі можливості для вчителів і учнів. Викладачі можуть використовувати готові курси як основу для занять, доповнюючи їх власними матеріалами. Домашні завдання можуть включати проходження окремих модулів на платформах, а практичні заняття — виконання інтерактивних завдань, які допомагають закріпити теоретичні знання. Залучення цифрових інструментів також стимулює учнів до активної участі в процесі навчання, адже вони можуть бачити реальні результати своєї роботи.

Водночас використання інтерактивних середовищ має і певні виклики. Наприклад, у школах із обмеженим доступом до інтернету чи комп'ютерної техніки виникають труднощі з впровадженням таких платформ у навчальний процес. Крім того, учні можуть потребувати додаткової мотивації для самостійного освоєння матеріалів, особливо якщо вони стикаються зі складними темами. Проте ці недоліки можна компенсувати завдяки комбінуванню інтерактивних платформ із традиційними методами викладання.

Отже, інтерактивні платформи є потужним інструментом для навчання веб-програмування. Їх використання сприяє розвитку ключових навичок і компетенцій, необхідних у сучасному цифровому світі. Для досягнення максимального ефекту викладачам важливо поєднувати роботу з платформами з індивідуальним підходом до учнів, забезпечуючи баланс між цифровими технологіями та живим спілкуванням.

### **2.1.2. Хмарні сервіси та онлайн-інструменти у викладанні веб-програмування.**

Хмарні сервіси та онлайн-інструменти у викладанні веб-програмування

Використання хмарних сервісів та онлайн-інструментів у навчанні веб-програмування дозволяє зробити процес навчання гнучкішим, доступнішим і сучаснішим. Ці технології не лише полегшують взаємодію між викладачем та учнями, а й відкривають широкі можливості для виконання практичних завдань у реальному часі без потреби в потужному технічному забезпеченні.

Хмарні сервіси, такі як Google Workspace (Docs, Sheets, Slides), активно застосовуються для організації колективної роботи та створення навчальних матеріалів. Наприклад, викладач може розробляти інтерактивні презентації чи чеклисти для перевірки знань, які учні можуть заповнювати онлайн. Для завдань із веб-програмування чудово підходить Google Sites — сервіс, що дозволяє створювати прості веб-сайти, використовуючи інтуїтивний інтерфейс. Учні, які роблять перші кроки у веб-розробці, можуть використовувати цей інструмент для базового ознайомлення зі структурою веб-сторінок. [11]

Середовище для написання коду у хмарі стало справжнім проривом у навчанні веб-програмування. Наприклад, платформи Replit, CodeSandbox та JSFiddle дозволяють працювати з HTML, CSS та JavaScript прямо в браузері, не потребуючи складного налаштування середовища на локальному комп'ютері. Учні можуть експериментувати з кодом, миттєво отримуючи результат, що дуже важливо для розуміння принципів роботи веб-технологій. Водночас викладачі можуть відстежувати прогрес, перевіряти завдання та надавати зворотний зв'язок у реальному часі.

Для організації навчального процесу важливо враховувати також хмарні системи керування навчанням (LMS), такі як Moodle, Google Classroom або Edmodo. Ці платформи дозволяють викладачам створювати курси, розміщувати навчальні матеріали, організовувати тестування та зберігати роботи учнів. Наприклад, викладач може підготувати окремий курс із основ

веб-програмування, який міститиме теоретичні матеріали, завдання з використанням хмарних інструментів та посилання на онлайн-ресурси.

Інструменти для спільної роботи над проектами також відіграють важливу роль. Сервіс GitHub, що широко використовується професіоналами, стає все популярнішим серед учнів. Його інтеграція у навчальний процес дозволяє навчити учнів основам роботи з системами контролю версій, спільного редагування коду та управління проектами. Інший схожий сервіс, GitLab, також можна використовувати для командних завдань і проєктів.

Хмарні сервіси надають ще одну важливу перевагу — вони полегшують організацію дистанційного навчання. У випадку обмежень, пов'язаних із пандемією чи іншими факторами, учні можуть повноцінно виконувати навчальні завдання, залишаючись удома. Наприклад, через Microsoft Teams або Zoom можна організувати заняття, використовуючи спільні онлайн-дошки (наприклад, Miro або Jamboard) для візуалізації матеріалів.

Однак важливо враховувати і певні виклики. Наприклад, доступ до таких інструментів залежить від якості інтернет-з'єднання, а також технічної підготовленості учнів і викладачів. Крім того, багато хмарних платформ мають платні функції, що може обмежувати їх використання в деяких навчальних закладах.

Підсумовуючи, хмарні сервіси та онлайн-інструменти є невід'ємною частиною викладання веб-програмування. Їх використання сприяє не лише підвищенню ефективності навчання, а й підготовці учнів до реальних умов роботи в ІТ-сфері, де такі технології є стандартом. Важливо лише забезпечити збалансований підхід, інтегруючи ці ресурси відповідно до потреб і можливостей як учнів, так і викладачів.

### **2.1.3. Використання візуалізацій та симуляторів у навчальному процесі**

Візуалізації та симулятори є потужними інструментами для викладання веб-програмування, оскільки вони дозволяють учням краще зрозуміти абстрактні концепції та взаємозв'язки між різними елементами веб-технологій.

Замість сухого вивчення теорії, використання графічних елементів і динамічних симуляцій допомагає зробити навчання більш захопливим і наочним.

Візуалізації, такі як графічні діаграми, схеми та інтерактивні презентації, сприяють кращому розумінню структури веб-сторінок, принципів роботи HTML, CSS і JavaScript. Наприклад, для пояснення каскадності стилів у CSS можна використовувати анімації, які демонструють, як стиль впливає на різні елементи сторінки. Інструменти на кшталт **CodePen** або JSFiddle дозволяють учням експериментувати з кодом і одразу бачити зміни в реальному часі, що є дуже важливим для вивчення роботи DOM (Document Object Model).

Симулятори додають навчальному процесу інтерактивності. Учні можуть використовувати спеціалізовані симулятори для тестування та налагодження свого коду без необхідності встановлення складних локальних середовищ. Наприклад, браузерні інструменти розробника (Chrome DevTools, Firefox Developer Tools) дозволяють у реальному часі редагувати код, аналізувати помилки та оптимізувати роботу веб-сторінки. [6] Викладачі можуть показувати приклади з використанням цих інструментів, щоб наочно продемонструвати, як знайти й виправити проблеми в коді.

Ще одним прикладом використання симуляторів є навчальні середовища для роботи з серверною частиною веб-програмування. Наприклад, віртуальні лабораторії, такі як Glitch або Heroku, дозволяють розгорнути сервери й тестувати взаємодію між клієнтом і сервером у реальному часі. Це особливо корисно для пояснення таких концепцій, як запити HTTP, обробка даних на сервері та робота з базами даних.

Для навчання основ алгоритмів і структур даних, які часто є невіддільною частиною веб-програмування, викладачі можуть використовувати платформи з візуалізацією коду, такі як VisuAlgo або Pythontutor. Вони дозволяють покроково показати, як працюють алгоритми, що допомагає краще зрозуміти їхню логіку. Наприклад, для пояснення роботи сортування чи пошуку в масивах ці інструменти демонструють, як змінюються дані на кожному кроці виконання алгоритму.

Візуалізації та симулятори також сприяють розумінню складних концепцій, таких як адаптивний дизайн або оптимізація веб-сторінок. Наприклад, використання симуляторів для тестування сторінок на різних пристроях (мобільних телефонах, планшетах, настільних комп'ютерах) допомагає учням зрозуміти, як розмітка й стилі впливають на відображення сайту. Інструменти на кшталт Responsive Design Mode у браузерях чи Webflow дозволяють наочно побачити результати роботи.

Не варто забувати й про можливості використання ігрових симуляторів у навчанні. Такі платформи, як Flexbox Froggy або Grid Garden, пропонують ігровий підхід до вивчення CSS Flexbox і CSS Grid. Цей формат заохочує учнів до активної участі в навчанні, адже завдання подаються у вигляді цікавих головоломок із поступовим ускладненням.

Однак, як і будь-які інші технології, візуалізації та симулятори мають свої обмеження. Вони не можуть повністю замінити живе спілкування з викладачем і потребують відповідного обладнання та доступу до інтернету. Але при правильному підході вони значно збагачують процес навчання, роблячи його сучасним і ефективним.

Підсумовуючи, використання візуалізацій та симуляторів у викладанні веб-програмування дозволяє перетворити складні концепції на доступні й цікаві. Вони допомагають учням краще розуміти матеріал, розвивати логічне мислення й застосовувати знання на практиці, що робить їх невід'ємною частиною сучасної освітньої методики.

#### **2.1.4. Роль технологій дистанційного навчання у формуванні навичок веб-програмування**

З розвитком інформаційних технологій дистанційне навчання стало важливою складовою освітнього процесу, включаючи вивчення веб-програмування. Зокрема, воно дозволяє створювати гнучкі й доступні умови для здобуття знань, що є важливим фактором у розвитку технологічної освіти в сучасному світі. Особливо це стосується таких динамічних і швидко

змінних галузей, як веб-розробка, де інструменти й технології постійно оновлюються.

Дистанційне навчання має ряд переваг, які роблять його ефективним інструментом для навчання веб-програмуванню. Зокрема, студенти мають можливість вивчати матеріал у зручний для себе час і в комфортному темпі. Це важливо, адже освоєння програмування потребує часу на самостійне розв'язання задач та експерименти з кодом, що набагато легше зробити в умовах гнучкого графіка.

Однією з головних переваг дистанційного навчання є можливість використовувати численні онлайн-платформи для вивчення веб-програмування. Наприклад, Codecademy, Udemy, freeCodeCamp та інші ресурси надають інтерактивні уроки, де учні можуть одразу практикувати написання коду. Вони надають доступ до великої кількості курсів, які охоплюють різні аспекти веб-розробки — від основ HTML і CSS до більш складних тем, таких як React або Node.js.

Інтерактивність є важливим елементом дистанційного навчання, оскільки вона дозволяє учням одразу бачити результати своїх дій. Це суттєво підвищує мотивацію та допомагає утримувати інтерес до навчання. Наприклад, на платформі freeCodeCamp студенти отримують можливість працювати з реальними проектами та завданнями, що дуже схоже на реальну веб-розробку, де кожен крок потребує пошуку рішень і вивчення нових підходів.

Крім того, у рамках дистанційного навчання велике значення має відео-лекції та вебінари. Такі онлайн-формати дають змогу викладачам донести складні концепції, пояснити принципи роботи технологій та продемонструвати приклади застосування цих знань на практиці. Викладач може проводити інтерактивні заняття, де студенти працюють разом над створенням веб-додатків або вирішенням технічних проблем. Це дозволяє побудувати інтерактивне навчальне середовище, яке сприяє не тільки теоретичному, а й практичному засвоєнню матеріалу.

Для вивчення веб-програмування у дистанційному навчанні також використовуються різноманітні інструменти для співпраці та обміну досвідом. Наприклад, платформами для спільної роботи можуть бути GitHub, де учні можуть завантажувати свої проєкти, працювати з версіями коду й обговорювати помилки, а також користуватися ресурсами для рецензування коду та перевірки результатів роботи. Це дозволяє створити досвід реального командного процесу розробки, що є важливою складовою навчання програмуванню.

Ще одним важливим інструментом є створення онлайн-курсів і міні-проєктів, які можуть бути виконані студентами в межах курсу. Веб-сайти на кшталт Coursera або edX пропонують інтерактивні навчальні модулі, де учні виконують завдання, тестують свої знання та отримують миттєвий фідбек. Це дозволяє не лише глибше зануритись у теми, а й відточити практичні навички, створюючи реальні веб-ресурси або застосунки.

Можливість працювати в онлайн-середовищах дозволяє студентам дистанційно знайомитися з різними фреймворками, бібліотеками й інструментами, які використовуються в професійній веб-розробці. Наприклад, Glitch і Repl.it дають змогу студентам розробляти та тестувати повноцінні веб-додатки без необхідності налаштовувати сервери та локальні середовища. Завдяки цьому студент може зосередитися на вивченні концепцій і логіки, не витрачаючи час на вирішення технічних питань налаштування програмного забезпечення.

У дистанційному навчанні також важливе місце займає зворотній зв'язок та індивідуальна підтримка студентів. Завдяки форумам, чатам і можливості комунікації через соціальні мережі чи спеціалізовані платформи студенти можуть отримати консультації з питань, які виникають під час виконання завдань. Це сприяє швидкому розв'язанню проблем і дозволяє розвивати навички самостійного навчання та пошуку рішень, що є важливою частиною розвитку професійних навичок програміста.

Не можна забувати і про важливість онлайн-сертифікацій, які надаються після успішного завершення курсів. Такі сертифікати можуть стати додатковим

стимулом для студентів, адже вони підтверджують їхні знання і навички перед потенційними роботодавцями. Наприклад, сертифікат з веб-програмування, отриманий через платформу Coursera, може стати важливим кроком для отримання практики або роботи в сфері технологій.

Таким чином, дистанційне навчання відіграє важливу роль у формуванні навичок веб-програмування. Воно дозволяє учням отримувати знання, працювати з реальними проектами, використовувати сучасні інструменти для розвитку своїх навичок і адаптуватися до постійно змінюваного технологічного середовища. З допомогою онлайн-курсів, інтерактивних платформ та інструментів для спільної роботи студенти можуть отримати практичний досвід і вдосконалювати свої навички веб-розробки, що є важливою складовою їхнього професійного розвитку.

#### **2.1.5. Інтеграція STEAM-підходу у викладання веб-програмування.**

В умовах сучасної освіти важливим є впровадження новітніх підходів, які сприяють розвитку критичного мислення, креативності та інноваційних навичок у учнів. Одним з таких підходів є STEAM — акронім, що позначає інтеграцію п'яти основних дисциплін: науки (Science), технології (Technology), інженерії (Engineering), мистецтва (Art) та математики (Mathematics). Такий підхід дозволяє створити цілісну освітню модель, у рамках якої учні не лише набувають теоретичних знань, але й застосовують їх на практиці для розв'язання реальних проблем.

У контексті веб-програмування, інтеграція STEAM-підходу є надзвичайно важливою. Програмування є важливою складовою технологій, але для ефективного навчання цьому важливо поєднувати технічні навички з творчими та аналітичними здібностями. Ось чому цей підхід дозволяє учням не тільки освоїти основи кодування, але й розвивати навички міждисциплінарної роботи, що є важливими для створення інноваційних веб-проектів.

Одним із способів впровадження STEAM-методів у навчання веб-програмуванню є включення елементів мистецтва та дизайну в навчальний

процес. Наприклад, під час навчання створенню веб-сайтів або інтерфейсів програм, учні не лише вивчають код HTML, CSS чи JavaScript, але й опановують принципи дизайну, колористики, типографії та користувацького досвіду (UX/UI). Це дозволяє їм отримати комплексне розуміння процесу розробки веб-продукту, включаючи як технічні, так і естетичні аспекти.

Викладання веб-програмування через призму STEAM також включає використання реальних кейсів та проєктного навчання. Наприклад, студенти можуть бути залучені до створення веб-додатків, які вирішують конкретні соціальні чи екологічні проблеми. Таким чином, вони застосовують знання з різних галузей науки, математики, техніки та мистецтва для створення функціональних і естетичних рішень, які мають реальний вплив. [16]

Використання цього підходу сприяє розвитку критичного мислення і здатності до розв'язання складних задач. Наприклад, у процесі навчання веб-програмуванню студенти часто стикаються з необхідністю оптимізації коду для забезпечення ефективності веб-ресурсів. Для цього необхідно враховувати не лише математичні та технічні аспекти, але й елементи дизайну та взаємодії з користувачем. Окрім цього, інтеграція інженерії у вивчення веб-розробки дозволяє учням згодом застосовувати навички кодування для створення більш складних веб-додатків або інтеграції різних технологій для вирішення певних задач.

Процес STEAM-навчання активно підтримується сучасними цифровими інструментами, які забезпечують учнів доступом до міждисциплінарних ресурсів. Наприклад, інтерактивні онлайн-платформи, такі як Tinkercad або Scratch, дозволяють учням створювати не тільки прості веб-сторінки, але й проєктувати 3D-моделі або програмувати анімації та ігри, що забезпечує розвиток вмінь в галузях, що не обмежуються лише програмуванням, але й технологічним дизайном та інженерією.

Варто зазначити, що такі платформи дозволяють студентам працювати не лише з чистим кодом, але й здійснювати візуалізацію результатів своєї роботи, що є важливим аспектом для розвитку креативних та дизайнерських навичок.

Крім того, інтеграція STEAM дає змогу організувати навчальні проєкти, в яких учні працюють у командах. Це допомагає розвивати навички співпраці та комунікації, що є важливими для майбутніх фахівців у будь-якій сфері, зокрема в веб-розробці.

Одним із практичних прикладів використання STEAM-підходу в веб-програмуванні може бути створення онлайн-платформ для вивчення природничих наук або історії, де учні можуть розробляти інтерактивні веб-сайти з елементами анімації, інтерактивних діаграм та карт. Такий проєкт не лише допомагає студентам освоїти веб-технології, але й розвивати навички роботи з даними, що є важливим аспектом майбутньої роботи в ІТ-сфері.

Інтеграція STEAM-підходу в освіту також дозволяє підвищити інтерес учнів до навчання, оскільки створює можливості для творчої самовираження і розв'язання реальних проблем через технології. Це особливо важливо у веб-програмуванні, де практичне застосування знань у вигляді готових проєктів допомагає учням краще розуміти як теорія перекладається в реальні продукти, що використовуються в повсякденному житті.

Впровадження STEAM-методів у навчальний процес дозволяє зробити освіту більш сучасною та цікавою, сприяючи розвитку важливих компетентностей, таких як критичне мислення, творчий підхід, вміння працювати в команді, а також здатність до інновацій. Такі підходи формують базу для успішної кар'єри в галузі веб-програмування та інших високотехнологічних індустрій.

#### **2.1.6. Інтерактивні методи навчання: гейміфікація, квести, вебінари.**

Сучасна педагогіка все більше орієнтується на інтерактивні методи навчання, які дозволяють не лише передати знання, а й активно залучати учнів до навчального процесу. У контексті викладання веб-програмування такі методи як гейміфікація, квести та вебінари, сприяють розвитку інтересу, мотивації та практичних навичок.

### **Гейміфікація в навчанні веб-програмування**

Гейміфікація — це використання ігрових елементів у неігрових контекстах, таких як освітній процес. Її мета — підвищення зацікавленості учнів через інтерактивність і змагальність. У веб-програмуванні гейміфікація може реалізовуватись наступними способами:

- Рівні та нагороди. Учні отримують «бали» за виконання завдань, розв'язання задач або створення проектів. Наприклад, завершення теми HTML може відкрити доступ до нових завдань із CSS.
- Таблиці лідерів. Ведення загального рейтингу за результатами уроків стимулює здорову конкуренцію серед учнів.
- Віртуальні сценарії. Створення ігрових сценаріїв, де учні мають розробити веб-сайт для вигаданого клієнта або вирішити проблему кодування.
- Ігрові платформи. Використання спеціальних платформ, таких як CodeCombat чи CSS Diner, які інтегрують навчання через гру, сприяючи легшому засвоєнню матеріалу.

Гейміфікація робить навчання цікавим і значущим, даючи учням можливість бачити конкретні результати своїх зусиль, що стимулює бажання навчатися далі.

### **Квести у навчанні веб-програмування**

Квести є ще одним захопливим методом інтерактивного навчання. Вони поєднують елементи гри, дослідження і вирішення проблем, стимулюючи учнів до пошуку нестандартних рішень.

- Структура квесту. Квест може включати низку завдань, які учні мають виконати, щоб досягти кінцевої мети. Наприклад, створення простого

веб-сайту з урахуванням усіх вимог замовника: від структури HTML до дизайну CSS.

- Елементи загадок. Учні можуть отримувати завдання у вигляді загадок або головоломок, відповіді на які допоможуть їм рухатися вперед у квесті. Наприклад, у коді може бути прихований підказковий текст, який потрібно знайти за допомогою інспектора елементів браузера.
- Робота в командах. Командні квести сприяють розвитку навичок співпраці. Наприклад, кожен член команди відповідає за певну частину проекту — структуру, дизайн або інтерактивність.

Квести особливо ефективні в мотивації учнів до навчання, адже вони дозволяють зануритися в ігровий сюжет, паралельно засвоюючи знання.

### **Вебінари як форма інтерактивного навчання**

Вебінари — це онлайн-семінари, які дозволяють проводити уроки в реальному часі за допомогою платформ, таких як Zoom, Google Meet чи Microsoft Teams. У навчанні веб-програмування вони є універсальним інструментом для передачі знань та взаємодії між учнями і викладачем.

Живе спілкування. Учні можуть задавати запитання в реальному часі, а викладач — миттєво пояснювати незрозумілі моменти.

Демонстрація коду. Викладач може показувати процес написання коду або виправлення помилок через спільний доступ до екрану.

Інтерактивні завдання. Під час вебінару можна організовувати короткі практичні завдання: наприклад, запропонувати учням створити базовий шаблон веб-сторінки за кілька хвилин і показати свої результати. [24]

Записи вебінарів. Важливим аспектом є можливість перегляду запису уроку, що дозволяє учням повторити матеріал у зручний час.

Вебінари також дозволяють запросити професіоналів із галузі веб-розробки, які можуть поділитися своїм досвідом, надихнувши учнів на освоєння нових технологій.

### **Комбінування методів**

Гейміфікація, квести та вебінари є взаємодоповнюючими методами. Наприклад, викладач може провести вебінар у формі квесту з елементами гейміфікації, де учні отримують бали за правильні відповіді чи виконання завдань. Такий підхід не лише підвищує залученість учнів, але й сприяє кращому засвоєнню складних тем.

У підсумку, інтерактивні методи навчання значно підвищують ефективність уроків веб-програмування, стимулюючи інтерес учнів і сприяючи розвитку їхніх технічних та креативних навичок. Використання таких методів також допомагає учням краще підготуватися до реальних викликів у сфері веб-розробки.

### **2.2. Проєктне навчання як основа для формування навичок веб-програмування.**

Проєктне навчання є одним із найефективніших підходів до розвитку навичок веб-програмування, оскільки воно інтегрує теоретичні знання з практичною діяльністю. Цей метод спрямований на виконання конкретних завдань, пов'язаних із реальними життєвими або професійними ситуаціями, що дозволяє учням відчувати себе частиною процесу розробки.

Проєктне навчання базується на активному підході до здобуття знань. Учні стають не лише пасивними споживачами інформації, а й активними учасниками навчального процесу. Наприклад, замість простого вивчення теоретичного матеріалу, вони створюють функціональні веб-сайти, в яких реалізують усі аспекти програмування — від верстки до інтеграції серверної логіки.

#### **Етапи проєктного навчання у веб-програмуванні**

1. Вибір теми проєкту. Учні можуть запропонувати власну ідею або обрати тему, запропоновану вчителем. Наприклад, створення веб-додатку для розкладу уроків або онлайн-каталогу книг.
2. Планування. На цьому етапі визначаються вимоги до проєкту, технології, які будуть використовуватися, і розподіляються ролі у команді (якщо проєкт груповий).
3. Реалізація. Учні працюють над створенням веб-продукту, поступово вирішуючи складніші завдання. Наприклад, спочатку створюється каркас сайту (HTML), потім додаються стилі (CSS), а на завершальному етапі — функціонал за допомогою JavaScript.
4. Презентація. Завершенням роботи над проєктом є його представлення аудиторії. Учні демонструють створений продукт, пояснюють використані рішення та труднощі, які вдалося подолати.
5. Оцінювання. Учитель оцінює результат проєкту, враховуючи як технічні аспекти (якість коду, функціональність сайту), так і навички командної роботи та презентації.

### Переваги проєктного навчання

Проєктний підхід дозволяє учням розвивати технічні та м'які навички одночасно. Працюючи над конкретними задачами, вони:

- освоюють сучасні інструменти веб-програмування, такі як Visual Studio Code, GitHub чи Bootstrap;
- вчаться шукати та використовувати додаткові ресурси, включаючи документацію та онлайн-курси;
- розвивають навички командної роботи, що є критично важливим для роботи у сфері ІТ.

Крім того, проєктне навчання сприяє формуванню критичного мислення. Учні стикаються з різними проблемами під час розробки і змушені шукати оптимальні рішення, аналізуючи та оцінюючи варіанти.

### **Застосування у профільній школі**

У профільних школах проєктне навчання може включати розробку комплексних проєктів, таких як багатосторінковий сайт із базами даних або інтерактивними компонентами. Наприклад, проєкт «Розробка онлайн-магазину» дозволить учням застосувати базові знання HTML і CSS, створити форму зворотного зв'язку через JavaScript та інтегрувати базу даних для збереження замовлень за допомогою PHP або Python. [28]

Дослідження від Edutoria підтверджують, що учні, залучені до проєктного навчання, демонструють значно вищі результати у розумінні предмета та у формуванні практичних навичок порівняно з тими, хто вивчає матеріал традиційними методами. [29]

Проєктний метод дозволяє підготувати учнів до реальних викликів у сфері ІТ, формуючи базові компетентності, необхідні для подальшого професійного розвитку.

### **2.3. Залучення реальних кейсів та проєктів для навчання веб-програмування.**

Використання реальних кейсів у навчанні веб-програмування сприяє розвитку практичних навичок, які необхідні для роботи в реальних умовах. У процесі роботи над такими завданнями учні краще розуміють сутність та значення веб-програмування. Крім того, це допомагає сформувати ключові компетенції, зокрема вміння розв'язувати комплексні завдання, критично мислити та взаємодіяти у команді.

Залучення реальних кейсів стимулює інтерес до навчання. Учні можуть створити веб-додаток або сайт для справжнього користувача — наприклад, для місцевої кав'ярні, школи чи благодійної організації. Це не лише допомагає

розвинути їхні навички, але й дозволяє зрозуміти, як веб-технології застосовуються в реальному житті.

Проектна робота забезпечує також ефективну інтеграцію теорії та практики. Наприклад, вивчення основ HTML і CSS можна закріпити через створення інформаційного веб-сайту, а вивчення JavaScript — через реалізацію інтерактивних функцій, таких як форми з валідацією.

Реальні кейси дозволяють вчителям моделювати умови, в яких працюють сучасні веб-розробники. Учні повинні враховувати реальні обмеження — часові рамки, технічні вимоги та побажання замовника. Завдяки цьому вони отримують досвід, який стане їм у пригоді в майбутньому працевлаштуванні.

Дослідження, проведені Code.org, підтверджують, що використання реальних кейсів сприяє підвищенню зацікавленості учнів у програмуванні та сприяє розвитку творчих здібностей. Водночас Web Design Library зазначає, що проектна робота стимулює командну взаємодію, яка є невіддільною частиною професії веб-розробника.

Методика включення реальних кейсів має бути ретельно продуманою. Важливо, щоб завдання були посильними для учнів, але водночас змушували їх виходити за межі комфортної зони, вирішуючи нетипові задачі. Наприклад, учитель може запропонувати створити багатосторінковий веб-сайт із використанням адаптивної верстки та мінімальної взаємодії на основі JavaScript. [21]

Успішно реалізований кейс приносить учням не лише знання, але й почуття задоволення від завершення реального проекту. Це підвищує їхню впевненість у власних силах і формує позитивне ставлення до веб-програмування як сфери діяльності.

#### **2.4. Розробка структури уроку веб-програмування (теоретична і практична частини).**

Структура уроку веб-програмування є важливою складовою ефективного навчального процесу, адже вона дозволяє чітко організувати час навчання,

забезпечити баланс між теоретичними знаннями та практичними навичками, а також адаптувати урок до рівня підготовки учнів. Веб-програмування як дисципліна поєднує як теоретичні аспекти, так і практичну реалізацію знань, тому при розробці уроку слід враховувати важливість кожної з цих частин.

**Теоретична частина уроку** має на меті забезпечити учнів базовими знаннями про основи веб-розробки, принципи роботи веб-технологій, а також надати їм теоретичну базу для подальшого застосування в реальних умовах.

Теоретичний матеріал включає:

- Вступ до теми уроку. На початку кожного уроку важливо визначити основну тему та її значення. Це дозволяє учням зрозуміти, навіщо вони вивчають ту чи іншу тему, і які знання отримають у кінці уроку. Важливо навести практичні приклади застосування технологій, що вивчаються, у реальному житті, що стимулює інтерес до навчання.
- Пояснення основних понять та принципів. Тут викладач має на меті дати чітке визначення ключових термінів і концепцій, таких як HTML, CSS, JavaScript, серверна та клієнтська частини веб-розробки. Важливо, щоб учні зрозуміли, як ці технології взаємодіють і як кожна з них впливає на результат — створення веб-сайтів та додатків.
- Пояснення теоретичних аспектів. Це може включати основи верстки, структурування веб-сторінок, принципи роботи з різними типами контенту, такими як текст, зображення, мультимедіа, а також робота з таблицями стилів (CSS) і динамічними елементами за допомогою JavaScript.
- Використання візуальних матеріалів. Для більш наочного засвоєння матеріалу викладач може використовувати презентації, відеоуроки, схемні зображення або інтерактивні ресурси, які демонструють, як працює той чи інший елемент на сторінці.
- Закріплення теоретичних знань. В кінці теоретичної частини уроку важливо організувати перевірку засвоєння учнями отриманих знань. Це можна зробити через короткі тести, усні опитування, обговорення

ключових питань або розбір задач, що дозволяють перевірити розуміння основних понять.

**Практична частина уроку** є не менш важливою, адже саме через практику учні закріплюють теоретичні знання та вчаться застосовувати їх на практиці. Практична частина уроку може включати:

- Розв'язання практичних завдань. Це може бути створення простих веб-сторінок, робота з тегами HTML, застосування стилів CSS для оформлення сторінок, додавання інтерактивних елементів за допомогою JavaScript. Практика має бути поступовою — від простих завдань до більш складних. [Додаток 1]
- Самостійна робота учнів. Одна з важливих складових практичної частини уроку — це самостійна робота учнів, де вони мають можливість застосувати на практиці набуті знання. Задачі повинні бути не надто складними, але водночас достатньо різноманітними, щоб кожен учень міг вибрати щось, що відповідає його рівню підготовки.
- Індивідуальна та групова робота. Часто веб-програмування передбачає командну роботу, тому учням можна запропонувати працювати над завданнями в групах. Це сприяє розвитку навичок співпраці та обміну ідеями. Використання групової роботи дозволяє швидше вирішувати складніші завдання, а також дає можливість учням ділитися досвідом.
- Демонстрація результатів. Після виконання практичних завдань важливо надати можливість учням продемонструвати свої результати. Це може бути, наприклад, демонстрація створених веб-сторінок або проектів. Такі заходи стимулюють учнів до кращого виконання завдань, адже вони бачать результат своєї роботи.
- Зворотний зв'язок. Оцінка результатів практичних завдань повинна бути не лише формальною, а й конструктивною. Викладач має надавати зворотний зв'язок, що допомагає учням зрозуміти, що саме вони зробили

правильно або де можуть покращити свої навички. Оцінювання не обов'язково має бути лише за допомогою балів — важливо, щоб воно було орієнтоване на підвищення мотивації та розвитку учнів.

- Підсумки уроку. В кінці уроку необхідно підвести підсумки, повторити основні моменти, розібрати проблеми, які виникли у процесі роботи, а також обговорити наступні кроки в навчанні. Такий підхід дозволяє не тільки закріпити знання, але й організувати своєчасну рефлексію з боку учнів.

У підсумку, структура уроку веб-програмування повинна включати чітке поєднання теоретичних та практичних частин. Теоретична частина повинна забезпечити учнів необхідними знаннями і розумінням основ веб-розробки, а практична — дозволити їм застосувати ці знання на практиці, створюючи реальні проєкти. Важливо, щоб уроки були побудовані так, щоб кожен учень мав можливість працювати як самостійно, так і в команді, отримуючи конструктивний зворотний зв'язок, що сприятиме його подальшому розвитку в сфері веб-програмування.

## **2.5. Аналіз ефективності розробленої методики**

### **2.5.1. Організація експериментального дослідження ефективності технологій навчання.**

Експериментальне дослідження є важливим етапом у визначенні ефективності нових підходів і технологій у навчанні веб-програмування. Це дозволяє порівняти традиційні методи навчання з сучасними технологічними підходами, оцінюючи їх вплив на рівень засвоєння матеріалу та розвиток практичних навичок учнів. Метою дослідження є вивчення того, як використання різних технологій, таких як інтерактивні платформи, онлайн-інструменти, проєктне навчання, впливає на ефективність навчання веб-програмування серед учнів старшої та профільної школи. [25]

Завдання дослідження включають оцінку рівня знань учнів до та після використання нових технологій, порівняння їх здатності до практичного застосування отриманих навичок, а також вивчення впливу різних методів на розвиток м'яких навичок, таких як критичне мислення та командна робота. Одним із головних завдань є також дослідження того, як інтерактивні методи навчання, такі як гейміфікація та вебінари, допомагають учням краще засвоювати складні концепції веб-програмування.

В експерименті беруть участь дві групи учнів: експериментальна і контрольна. У експериментальній групі використовуються новітні методи навчання, зокрема інтерактивні платформи та онлайн-сервіси для програмування, а також проектне навчання. Контрольна група навчається за традиційними методами, що включають використання підручників, лекцій і практичних завдань без інтерактивних компонентів.

Збір даних здійснюється через аналіз результатів тестувань, спостереження за процесом навчання та аналіз проектів, виконаних учнями. Важливою частиною дослідження є оцінка впливу нових технологій на здатність учнів застосовувати отримані знання на практиці, а також на їх мотивацію до навчання. Зокрема, проводиться порівняння якості проектів, виконаних учнями з різних груп, а також оцінюються м'які навички, які вони розвивають, працюючи з новими технологіями, такими як співпраця, креативність і вміння працювати в команді.

Очікується, що результати дослідження підтвердять гіпотезу про те, що використання інтерактивних технологій і проектних методів навчання покращить як засвоєння теоретичного матеріалу, так і практичних навичок з веб-програмування. Учні, які працюють з реальними кейсами, використовуючи сучасні онлайн-інструменти, матимуть кращі результати в порівнянні з учнями, які навчаються за традиційною програмою. [27]

Для оцінки ефективності дослідження використовуються різноманітні методи збору даних, такі як аналіз результатів тестів, опитування учнів і вчителів, спостереження за навчальним процесом. Такий підхід дозволяє

отримати об'єктивну інформацію про вплив нових технологій на якість навчання.

### **2.5.2. Оцінка рівня сформованості знань і навичок у старшокласників.**

Оцінка результатів дослідження є важливою частиною експериментальної роботи, оскільки дозволяє визначити ефективність використаних методів та технологій навчання веб-програмування. Після проведення експерименту важливо провести комплексну оцінку якості навчальних досягнень учнів, визначити, наскільки нові технології допомогли в засвоєнні матеріалу та якість набуття практичних навичок.

Оцінка проводиться за кількома критеріями. По-перше, важливо вивчити рівень знань учнів до і після впровадження нових технологій. Це може бути здійснено через тестування, де оцінюється не лише загальний рівень знань, а й уміння учнів вирішувати практичні завдання. За допомогою тестів можна порівняти прогрес у розумінні веб-програмування, оцінити здатність учнів застосовувати знання для розв'язання реальних задач. За результатами тестувань учнів, що використовували інтерактивні платформи, зокрема онлайн-інструменти для програмування, зазвичай спостерігається значний приріст знань, порівняно з традиційними методами навчання.

По-друге, важливо оцінити практичні досягнення учнів. Для цього застосовуються проєктні завдання, де учні мають змогу продемонструвати свої навички у створенні реальних веб-сайтів чи додатків. Завдяки використанню сучасних інструментів веб-розробки (наприклад, HTML, CSS, JavaScript), учні отримують можливість не лише освоїти теоретичні аспекти, а й втілювати свої ідеї у реальних проєктах. Це дозволяє порівняти, наскільки учні можуть успішно застосовувати знання на практиці і як вони здатні працювати в команді, вирішуючи комплексні задачі.

Одним із важливих аспектів є також оцінка розвитку м'яких навичок учнів. Робота над проєктами, взаємодія в команді, розв'язання конкретних завдань на реальних кейсах сприяють розвитку таких важливих компетенцій, як

критичне мислення, здатність до вирішення проблем, ефективна комунікація. Це має позитивний вплив на загальний розвиток учнів, а також на їхню готовність до подальшого навчання та працевлаштування в галузі ІТ. Зокрема, учні, які працюють у командах над реальними проектами, розвивають навички управління проектами, що є корисним не лише в контексті веб-програмування, а й в інших професійних сферах.

Загалом, дослідження показують, що застосування нових технологій у навчанні веб-програмування позитивно впливає на розвиток учнів. Вони не лише покращують технічні навички, а й здобувають необхідні для сучасного ринку праці м'які навички, які є важливими для успішної кар'єри в галузі інформаційних технологій.

### **2.5.3. Порівняльний аналіз результатів до і після впровадження методики.**

У даному розділі проводиться порівняльний аналіз результатів навчання учнів до і після впровадження розробленої методики веб-програмування. Метою цього аналізу є оцінка ефективності застосованих методів та інструментів, а також вивчення їх впливу на рівень знань та навичок учнів, а також на їх мотивацію до навчання.

До впровадження нової методики навчання веб-програмуванню учні здебільшого працювали за традиційними підходами, де основний акцент робився на теоретичних знаннях. Часто уроки не включали достатньої практичної складової, що обмежувало можливості учнів для поглибленого засвоєння матеріалу. Унаслідок цього учні часто не мали змоги реалізувати набуті знання на практиці, що негативно позначалося на їх мотивації та результатах навчання. [15]

Після впровадження нових методик, зокрема таких, як використання сучасних онлайн-платформ, інтерактивних інструментів, проєктного навчання і гейміфікації, спостерігаються значні зміни в навчальному процесі. Однією з найбільших переваг стало підвищення інтересу учнів до веб-програмування, оскільки вони мали змогу працювати з реальними проєктами, що допомогло

краще зрозуміти теоретичні аспекти та застосовувати їх у практиці. Важливим фактором стало і використання хмарних технологій, які дозволили учням працювати з проектами в реальному часі, незалежно від місця їх знаходження.

Результати аналізу показали, що після впровадження нової методики рівень засвоєння матеріалу учнями значно зріс. Як підтвердження, можна навести результати тестувань і практичних завдань, що дозволяють порівняти прогрес учнів до і після змін у методиці. Учні, які брали участь у проєктному навчанні, демонстрували набагато кращі результати як у теоретичних, так і в практичних завданнях. Крім того, учні стали більш самостійними в розв'язанні задач, швидше і точніше знаходили рішення, а також значно зросла кількість успішно завершених веб-проєктів.

Проєктне навчання, зокрема, стало важливим аспектом у зміні підходів до викладання. У порівнянні з традиційними методами, де учні здебільшого працювали індивідуально, нові методики навчання заохочували колективну роботу. Командні завдання дозволяли не лише вдосконалити технічні навички, але й розвивати комунікативні здібності учнів, їх здатність до співпраці та управління проєктами. Такий підхід також сприяв більш ефективному засвоєнню матеріалу, оскільки учні могли отримувати зворотний зв'язок від своїх одногрупників та викладачів.

Водночас, впровадження нових технологій та методів навчання не обійшлося без труднощів. Одним з викликів стала потреба в забезпеченні учнів доступом до відповідного обладнання та інтернет-ресурсів, особливо в сільських школах або у навчальних закладах, де фінансування на модернізацію технологічної бази було обмежене. Також деякі учні зіткнулися з труднощами адаптації до нових інструментів, оскільки вони мали обмежений досвід роботи з подібними технологіями.

Загалом, порівняльний аналіз результатів до і після впровадження нових методик показав, що зміни у підходах до викладання веб-програмування призвели до значного покращення результатів учнів. Зокрема, підвищення рівня теоретичних знань та практичних навичок, збільшення мотивації до навчання та

розвиток навичок командної роботи стали основними позитивними змінами в навчальному процесі.

#### **2.5.4. Вплив сучасних технологій на мотивацію та компетентності учнів.**

Сучасні технології мають значний вплив на процес навчання, зокрема в таких галузях, як веб-програмування. Вони змінюють традиційні підходи до навчання, роблячи його більш інтерактивним, доступним і персоналізованим. Вплив новітніх інструментів та платформ на мотивацію учнів важко переоцінити. Веб-ресурси, мобільні додатки, хмарні сервіси, інтерактивні навчальні платформи дозволяють створювати навчальні середовища, які не тільки стимулюють інтерес до навчання, але й допомагають учням розвивати необхідні навички та компетентності, важливі для успіху в сучасному світі. [19]

Одним з основних ефектів впровадження сучасних технологій є підвищення мотивації учнів до навчання. Це досягається завдяки інтерактивним інструментам, які роблять навчальний процес цікавим і динамічним. Наприклад, гейміфікація, використання квестів, інтерактивних вебінарів, а також робота з реальними проєктами створюють умови для більш глибокого залучення учнів у навчання. Завдяки використанню таких методів учні починають відчувати себе частиною реального процесу, що підвищує їхню зацікавленість та бажання досягти успіху в навчанні.

Веб-програмування, як одна з основних галузей, де застосовуються новітні технології, надає учням можливість практично застосовувати свої знання. Це особливо важливо, оскільки практичні навички стають вирішальними у розвитку компетентностей. Веб-розробка вимагає від учнів не тільки теоретичних знань, але й умінь, пов'язаних з плануванням, дизайном, програмуванням, тестуванням та розгортанням веб-сайтів і додатків. Завдяки онлайн-ресурсам і платформам учні можуть самостійно реалізовувати власні проєкти, що дозволяє їм розвивати навички програмування і логічного мислення.

Завдяки інструментам, які забезпечують доступ до навчальних матеріалів та інструкцій у будь-який час і з будь-якого місця, учні мають змогу самостійно регулювати свій навчальний процес. Це дозволяє не лише підвищити рівень їхньої компетентності, але й формує у них критичне мислення, здатність до самонавчання та адаптації до нових умов. Такі навички є важливими не тільки для розвитку в галузі веб-програмування, але й для майбутньої професійної діяльності в інших сферах.

Також варто зазначити, що інтеграція сучасних технологій у навчальний процес сприяє розвитку креативних і комунікативних навичок учнів. Розробка веб-проектів передбачає використання різних засобів, таких як графічний дизайн, створення інтерактивних елементів і анімацій, що вимагає креативного підходу. Одночасно учні набувають навичок співпраці у команді, обміну ідеями та вирішення проблем. Це допомагає їм стати більш гнучкими та адаптивними до змін у навчальному середовищі та в реальному світі.

Не менш важливим є і вплив технологій на розвиток комп'ютерної грамотності учнів. Сучасні інструменти допомагають формувати в учнів базові та поглиблені навички роботи з комп'ютерами, що є необхідним для їх подальшої професійної діяльності. Вони отримують навички програмування, роботи з базами даних, створення веб-сайтів і мобільних додатків, а також знайомляться з найсучаснішими технологіями в цій галузі, що значно підвищує їх конкурентоспроможність на ринку праці. [29]

Таким чином, сучасні технології мають комплексний вплив на мотивацію та компетентності учнів. Вони дозволяють створювати більш гнучкі та доступні умови для навчання, стимулюють учнів до самостійної роботи та розвитку практичних навичок, що допомагає їм досягати високих результатів у вивченні веб-програмування та в інших дисциплінах.

## ВИСНОВКИ

### 1. ОСНОВНІ ВИСНОВКИ ДОСЛІДЖЕННЯ

Веб-програмування стає невід'ємною частиною сучасної освітньої програми, і його значення для формування ключових компетентностей учнів важко переоцінити. Процес навчання веб-програмуванню має значний вплив на розвиток учнів у багатьох аспектах: від формування технічних навичок до розвитку креативності, критичного мислення та здатності працювати в команді.

Одним із ключових висновків є те, що інтеграція сучасних технологій у процес навчання сприяє значному покращенню мотивації учнів. Використання інтерактивних платформ, гейміфікації, онлайн-курсів та практичних проєктів дозволяє учням не тільки краще засвоювати матеріал, але й відчувати більшу залученість у навчальний процес. Сучасні технології роблять навчання більш доступним та цікавим, що є важливим чинником у розвитку навчальних мотивацій.

Завдяки використанню хмарних сервісів та онлайн-інструментів, учні мають змогу працювати над проєктами в реальному часі, отримуючи зворотний зв'язок від викладачів та однокласників, що сприяє їхньому розвитку в командних проєктах та зміцненню практичних навичок. Крім того, хмарні технології забезпечують учням доступ до навчальних матеріалів з будь-якої точки світу, що робить процес навчання гнучким і зручним для всіх учасників.

Проєктне навчання, як один із основних методів у викладанні веб-програмування, доводить свою ефективність у формуванні практичних навичок. Робота з реальними кейсами та створення власних веб-проєктів дозволяє учням застосовувати отримані знання в реальних умовах, а це є важливим етапом у їхньому професійному становленні. Такі методи навчання сприяють розвитку в учнів не тільки технічних знань, а й розвивають критичне мислення, здатність до вирішення проблем, командну роботу та креативний підхід до завдань.

Використання технологій дистанційного навчання та інтеграція STEAM-підходу у навчальний процес також позитивно впливають на розвиток учнів. Застосування таких методів дозволяє не лише поглиблювати знання в галузі веб-програмування, але й формує у учнів здатність самостійно знаходити рішення, що є важливим аспектом у їхньому особистісному та професійному розвитку.

Оцінка ефективності запропонованої методики навчання веб-програмуванню на основі експериментальних досліджень показала позитивні результати. Впровадження інтерактивних методів навчання, проєктного підходу та використання сучасних технологій призводить до покращення успішності учнів, їхньої зацікавленості в навчанні та розвитку практичних навичок.

У результаті дослідження можна зробити висновок, що застосування сучасних технологій у викладанні веб-програмування є важливим і необхідним кроком для розвитку системи освіти. Це дозволяє створити умови для більш ефективного і гнучкого навчання, підвищує зацікавленість учнів та допомагає їм розвивати важливі навички, необхідні для успішної кар'єри в інформаційних технологіях.

## 2. РЕКОМЕНДАЦІЇ ДЛЯ ВПРОВАДЖЕННЯ У НАВЧАЛЬНИЙ ПРОЦЕС

1. **Інтеграція сучасних онлайн-платформ та хмарних сервісів.** Одним із ефективних кроків для покращення навчального процесу є активне використання онлайн-платформ та хмарних сервісів. Це дозволить учням отримувати доступ до матеріалів та завдань будь-коли і з будь-якого місця, а також дасть змогу їм взаємодіяти з однокласниками та викладачами в реальному часі. Зокрема, платформи, як GitHub, GitLab для хостингу проєктів і кодів, а також сервіси для створення онлайн-курсу на платформі Moodle чи Google Classroom, можуть значно спростити процес навчання.

2. **Використання гейміфікації та інтерактивних методів.** З метою підвищення мотивації та залучення учнів до навчання, важливо застосовувати елементи гейміфікації. Це може бути впровадження онлайн-ігор, квізів, інтерактивних завдань, в яких учні зможуть отримувати бали за виконання завдань, порівнювати свої досягнення з іншими учасниками та отримувати зворотний зв'язок. Такий підхід робить процес навчання більш цікавим і захоплюючим.
3. **Активне використання проєктного навчання.** Використання реальних кейсів та проєктів є одним з найбільш ефективних способів навчання веб-програмуванню. Рекомендується створювати проєкти, де учні повинні розробити повноцінні веб-сайти або веб-додатки, що мають практичне значення. Це дозволить не тільки отримати теоретичні знання, але й застосувати їх на практиці. Проєктне навчання також сприяє розвитку навичок командної роботи, планування та управління часом.
4. **Впровадження технологій дистанційного навчання.** Для забезпечення гнучкості навчання та адаптації до різних умов навчання необхідно впровадити технології дистанційного навчання. Це дозволить учням вивчати матеріали у зручному для них темпі, отримувати зворотний зв'язок від викладачів і виконувати завдання з будь-якого місця. Системи дистанційного навчання (Google Classroom, Moodle) дозволяють учням мати доступ до навчальних матеріалів, а також виконувати практичні завдання, що є важливими для освоєння веб-програмування.
5. **Інтеграція STEAM-підходу.** Враховуючи важливість інтердисциплінарного підходу до навчання, доцільно впроваджувати STEAM-підхід у викладання веб-програмування. Це допоможе учням бачити зв'язок між програмуванням і іншими науками, такими як математика, фізика, мистецтво та інженерія. Застосування STEAM-підходу дозволяє формувати у учнів не тільки технічні навички, але й критичне мислення, креативність та здатність вирішувати складні завдання.

6. **Оцінювання на основі практичних результатів.** Традиційні методи оцінювання знань часто не дають повної картини про рівень підготовки учня у сфері веб-програмування. Тому рекомендується впроваджувати оцінювання на основі виконаних проєктів і практичних завдань. Це дозволить не лише визначити рівень знань учня, але й оцінити його здатність застосовувати отримані знання у реальних ситуаціях. Оцінювання за допомогою реальних проєктів сприяє глибшому розумінню предмету.
7. **Розвиток творчих та креативних навичок.** Важливо включати в навчальний процес елементи, що сприяють розвитку творчих та креативних навичок. Це можна досягти, надаючи учням можливість працювати над унікальними проєктами, де вони можуть застосовувати свої ідеї та розвивати інтерфейси, що є важливою складовою веб-програмування. Наприклад, створення веб-додатків для вирішення конкретних проблем чи розробка інтуїтивно зрозумілих інтерфейсів.
8. **Залучення зовнішніх спеціалістів та індустріальних експертів.** Для того, щоб учні могли ознайомитися з реальними вимогами ринку та тенденціями в веб-програмуванні, рекомендується запрошувати зовнішніх експертів, програмістів та розробників. Вони можуть провести майстер-класи, онлайн-лекції або консультації для учнів, що дозволить поглибити їхнє розуміння галузі, ознайомити з сучасними інструментами та методами роботи в реальному середовищі.

Ці рекомендації є важливими кроками для удосконалення навчання веб-програмуванню в школах. Впровадження таких підходів сприятиме не лише підвищенню якості знань учнів, а й розвитку важливих навичок, необхідних для успішної кар'єри в галузі інформаційних технологій.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Методика навчання інформатики [Електронний ресурс] / Львівський національний університет імені Івана Франка. URL: <https://pedagogy.lnu.edu.ua/wp-content/uploads/2016/11/Навчальний-посібник-1.pdf>
2. Навчальний посібник. Шкільний курс інформатики та методика його викладання. URL: <http://dspace.pnpu.edu.ua/bitstream/123456789/3982/1/barbolina1.pdf>
3. Що таке Інтернет? Як працює Інтернет. – URL: <https://searchwindevelopment.techtarget.com/definition/Internet>
4. Рекомендації товарів на сайті – URL: <https://esputnik.com/blog/tovarnye-rekomendacii-na-sajte-instrumentpovysheniya-konversii>
5. Трошкі Н.В. Робоча програма навчальної дисципліни викладання інформатики в школі. URL: <https://www.uzhnu.edu.ua/en/infocentre/get/44631>
6. Грокаємо алгоритми – Адітья Бхаргава, 2017. – 290 с.
7. Веб-додаток – вільна енциклопедія. – URL: <https://www.maxcdn.com/one/visual-glossary/web-application/>
8. Розробка веб-додатків. – URL: <https://www.comentum.com/guideto-web-application-development.html>
9. Visual Studio Code. – URL: <https://blog.eduonix.com/softwaredevelopment/visual-studio-code-popular/>
10. Інженер програмного забезпечення вивчає HTML5, JavaScript і jQuery: посібник для веб-додатків, заснованих на стандартах / Дейн Кемерон, 2013. – 257 с.
11. Респонсивний веб-дизайн з HTML5 та CSS: розробка веб-сайтів, які не зламаються на майбутньому, за допомогою новітніх технологій HTML5 і CSS, Бен Фрейн, 2020. – 410 с.

12. HTML: мова розмітки гіпертексту. – URL:  
<https://developer.mozilla.org/en-US/docs/Web/HTML>
13. Таблиці каскадних стилів CSS. – URL: <http://htmlbook.ru/css>
14. Що таке JavaScript? Повний стек програмування. – URL:  
<https://www.infoworld.com/article/3441178/what-is-javascript-the-full-stack-programming-language.html>
15. Що таке Bootstrap? Кращі практики фреймворку. – URL:  
<https://wpamelia.com/what-is-bootstrap/>
16. 15 найкращих фреймворків CSS для розробників у 2020 році. – URL:  
<https://www.mockplus.com/blog/post/css-framework>
17. Введення до мов програмування для бекенду. – URL:  
<https://www.educba.com/back-end-programming-languages/>
18. Чому Laravel є найкращим PHP-фреймворком для використання в 2020 році? – URL:  
<https://www.freecodecamp.org/news/why-laravel-is-the-best-php-framework-to-use-in-2020/>
19. Наказ Міністерства освіти і науки України № 614 від 21.06.2010 "Про затвердження вимог до специфікації навчального комп'ютерного комплексу для кабінетів інформатики та інформаційно-комунікаційних технологій навчальних закладів системи загальної середньої освіти". URL: <https://zakon.rada.gov.ua/rada/show/v0614290-10>.
20. Канівець, М. Інноваційна технологія «портфолію» як інструмент формування готовності студентів до професійно особистісного саморозвитку / Канівець Марина // Інновації в освіті : Міжнар. наук.- 306 метод. конф., 16–17 жовт. 2012 р., м. Київ : тези доп. / Київ. нац. торг.-екон. ун-т. – Київ : КНТЕУ, 2012. – С. 279–281.
21. Що таке MySQL? Визначення, функції, пояснення. – URL:  
<https://blog.templatetoaster.com/what-is-mysql/>
22. Що таке MVC? Переваги та недоліки MVC. – URL:  
<https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/>

23. Open Server – портативна серверна платформа та програмне середовище.  
– URL: <https://alternativeto.net/software/open-server/>
24. Плюси і мінуси програмування на фреймворку Vue.js. – URL:  
<https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>
25. Стрілецька Н. М. Методика навчання інформатики (у початковій школі):  
навчально-методичний посібник / Н. М. Стрілецька. – Чернігів:  
Чернігівський національний педагогічний університет імені Т. Г.  
Шевченка, 2014. – 240 с.
26. Використання освітніх онлайн-матеріалів: сучасні підходи і технології  
нової української школи / Державна установа "Український інститут  
розвитку освіти". URL:  
<https://uied.org.ua/wp-content/uploads/2023/03/metodychni-rekomendacziyi.pdf>
27. Карабін О.Й., Петрів Х.Б. Формування основ вебдизайну в учнів старших  
класів закладів загальної середньої освіти. URL:  
[http://www.innovpedagogy.od.ua/archives/2022/53/part\\_1/9.pdf](http://www.innovpedagogy.od.ua/archives/2022/53/part_1/9.pdf)
28. Методичні рекомендації щодо викладання інформатики у 5-11 класах у  
2022/2023 навчальному році [Електронний ресурс] / Міністерство освіти і  
науки України. URL:  
[https://znavshov.com/FR/16401/13-15\\_684-686-lipen-serpen-2022-5-8.pdf](https://znavshov.com/FR/16401/13-15_684-686-lipen-serpen-2022-5-8.pdf)
29. Савчук Т. Гейміфікація у викладанні веб-програмування. URL:  
<https://lib.iitta.gov.ua/id/eprint/601234/>
30. Глибовець М.М., Завадський І.О. Вступ до вебтехнологій. Посібник для  
10–12 класів закладів загальної середньої освіти та вищих навчальних  
закладів. URL:
31. Морзе Н.В. Інформатика: підруч. для 10 кл. загальноосвітніх навчальних  
закладів / Н.В. Морзе, В.П. Вембер, О.Г. Кузьмінська.
32. Євтеєв В. М., Семеріков С. О., Теплицький І. О. Досвід вивчення  
інтерактивних Web-технологій в середній школі та педагогічному ВНЗ.  
URL:

[https://elibrary.kdpu.edu.ua/bitstream/0564/783/1/Evteev\\_Semerikov\\_Teplitskiy.pdf](https://elibrary.kdpu.edu.ua/bitstream/0564/783/1/Evteev_Semerikov_Teplitskiy.pdf)

## ДОДАТОК 1

### План-конспект уроку з інформатики

**Тема уроку:** Розв'язання практичних завдань із веб-програмування

**Мета уроку:**

- Навчальна: ознайомити учнів із основами створення веб-сторінок, використанням HTML, CSS та JavaScript.
- Розвивальна: розвивати логічне мислення, уважність до деталей та творчі здібності.
- Виховна: формувати в учнів наполегливість, відповідальність за результат роботи.

### Обладнання:

- Комп'ютери з доступом до мережі Інтернет.
- Програмне забезпечення: текстовий редактор (Visual Studio Code, Notepad++ тощо), сучасний веб-браузер.
- Методичні матеріали (приклади HTML-коду).

**Тип уроку:** комбінований (вивчення нового матеріалу та практична робота).

### Структура уроку:

1. Організаційний момент (3 хв).
2. Актуалізація знань (5 хв).
3. Мотивація навчальної діяльності (5 хв).
4. Вивчення нового матеріалу (15 хв).
5. Практична робота (15 хв).
6. Підбиття підсумків (5 хв).
7. Домашнє завдання (2 хв).

### Хід уроку

### **1. Організаційний момент (3 хв)**

- Перевірка присутності учнів.
- Перевірка готовності робочих місць.

### **2. Актуалізація знань (5 хв)**

Фронтальне опитування:

- Що таке веб-сторінка?
- Які теги HTML ви знаєте?
- Для чого використовуються CSS і JavaScript?

### **3. Мотивація навчальної діяльності (5 хв)**

- Обговорення реальних прикладів: «Як створюються сайти, які ми використовуємо щодня?»
- Демонстрація простої веб-сторінки з базовим дизайном.

### **4. Вивчення нового матеріалу (15 хв)**

#### 1. HTML-структура веб-сторінки:

- Теги `<html>`, `<head>`, `<body>`.
- Додавання заголовків, абзаців, списків.

#### 2. CSS для оформлення:

- Як підключити стилі до HTML-документа.
- Основні властивості стилів: колір, розмір тексту, фон.

#### 3. JavaScript:

- Додавання інтерактивності (наприклад, створення кнопки, що змінює текст).

### **5. Практична робота (15 хв)**

Завдання 1. Створення простої веб-сторінки

1. Створити HTML-документ із базовою структурою.
2. Додати заголовок, абзац і список із 3 пунктів.

#### Завдання 2. Робота з CSS

1. Змінити колір фону сторінки.
2. Додати стилі для списку (змінити колір тексту, стиль маркерів).

#### Завдання 3. Додавання JavaScript

1. Додати кнопку, яка при натисканні змінює текст на сторінці.

### **6. Підбиття підсумків (5 хв)**

- Обговорення результатів роботи.
- Відповіді на запитання учнів.
- Оцінювання активності учнів та їхніх результатів.

### **7. Домашнє завдання (2 хв)**

- Завершити завдання, розпочаті на уроці (за необхідності).
- Прочитати додаткові матеріали про властивості CSS та події в JavaScript.

**Примітка:** Усі роботи зберегти на комп'ютері для подальшого аналізу.